# Scalable Programming for the Analysis of Aphasia Transcripts

Paula Garcia
Rochester Institute of Technology
Computing and Information Sciences
pxg5962@rit.edu

Vicki Hanson
Rochester Institute of Technology
Computing and Information Sciences
vlhics@rit.edu

## ABSTRACT

Access technologies designed for individuals with non-fluent aphasia focus on digitizing speech therapy methods and generating speech. To improve these technologies, the language characteristics of individuals with non-fluent aphasia must be further understood. Language corpuses, such as the AphasiaBank, provide a promising solution for informing technology usability in terms of navigation, interface, and content decisions. As a tool for informing such work, this research investigates the viability of a flexible and scalable multi-threaded software program for the analysis of AphasiaBank transcripts. Results show that the program allows rapid analysis of all transcriptions by optimizing core functionality and minimizing the number of areas for synchronization. This research aims to improve the access to information, products, and services in technology for individuals with non-fluent aphasia.

## INTRODUCTION

More than 1 million stroke and head injury survivors in the United States have Aphasia. Non-fluent aphasia is one of the most common types of Aphasia, effecting ones' ability to produce language syntax and understand written text [3]. Individuals with non-fluent aphasia however, maintain comprehension of spoken language and phonological production. Researchers from varying disciplines including speech pathology, linguistics, and neuroscience have investigated how both language attributes (phonology and syntax) may be used to regain spoken language fluency [1, 3]. Meanwhile, technology and appliances that rely on textual comprehension and spoken fluency are largely inaccessible to individuals with non-fluent aphasia [7].


Existing software for text categorization includes N-Gram, Wordle, DocuBurst, among others [2,8]. Although these tools provide insights towards the frequency and associations of words, they do not provide in depth information regarding the phonology and syntax of language in aphasia. Spoken language corpora present promising opportunities for supporting accessibility in communication by providing comparable texts among a set of users/participants. This study tests the feasibility of a customized, scalable transcript analysis of individuals with aphasia, in order to provide actionable insights for creators of accessible technologies.

## METHODS

The program works as follows:

- $S_1$: File manager locates all available files and stores them in an array.
- $S_2$: Initiates threads based on the number of files and processors available. Each thread is provided a copy of all the files with a range to read.
- $S_3$: Thread reads the files and tries to find the number of syllables for each word. When all words have been processed, it updates all participants' information.
- $S_4$: Calculates the number of syllables and creates an object of $S_5$.
- $S_5$: Stores various attributes for a word (i.e syllable count, location, length of sentence, etc.).

The efficacy of the program was tested to confirm scalability and time-efficiency. The program should support larger transcript volumes and additional linguistic testing functions. First, eighty-eight transcripts were tested for thread processing time on strings versus threads. Referring to figure one, this would mean that $S_1$ would read the files, store the strings and send the strings to $S_2$ for processing. When sending over files, the average processing time was 0.25 seconds. When sending over strings for processing, the average completion time was 1.50 seconds. For this reason, testing continued on the program that used files for transferring, as initially explained in Figure one.

Next, duplicate files were randomly generated from the language corpus to simulate the processing of a larger set of files. The goal was to test the efficiency of the program with multiple threads. This bootstrapped dataset is a simulated representation of processing times, providing approximate processing times at each scale.

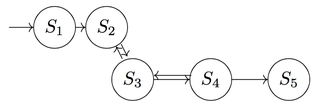



Figure 1: Graph of the program classes and their interactions.

**Testing Environment Specifications**

| | Details |
|---|---|
| Model | iMac, late 2012 |
| Processor | 2.9 GHz Intel Core i5 |
| Memory | 32 GB 1333 MHz DDR3 |
| OS | OSX Yosemite v. 10.10.3 |

Table 1: Specifications for analysis

**Simulated File Set**

| File Count | Total Size | Number of Lines |
|---|---|---|
| 500 | 4.5 MB | 85,217 |
| 750 | 6.4 MB | 123,507 |
| 1,000 | 7.9 MB | 158,479 |
| 1,500 | 12.7 MB | 237,472 |
| 1,750 | 16.2 MB | 294,948 |

Table 2: Simulated File Set for Program Testing

## RESULTS

The program was run on various data sizes, as shown in table five. Table six includes the completion time for each set of file sizes and number of cores. Generally, more files can be processed in a shorter period of time as the number of threads increases. The only exceptions are the completion times for three and four threads on 1,750 files. The number of files was divided in this manner because no more than 1,800 files could be processed with the computers given memory. For three and four threads, the completion times are 19.14 and 19.54 where two threads is 18.49 seconds. This does not follow the trend from the previous completion times.

The interview parse times were evaluated to see if there was a significant difference in completion times. Each thread was able to parse a transcript (removing single-letter words and symbols) in under a second. The next possible explanation could be that since the text was being split for each thread by the number of lines, some threads may process more words than others. The counts for each word processing per thread are included below.

As shown, the threads had at maximum, a +-10 word difference. Taking these results into account, the next area to investigate is the influence of memory on processing time. A delay in processing may be attributed to the fact that 1,750 files were approaching the memory processing limits of the machine. In other words, the processing times may become less efficient as the computer reaches its peak memory allocation.

The patterns for completion time and their changes can be seen on the plot in Figure two. Overall, the program shows the potential to be scalable as the number of files increases. However more exploration is needed to verify that the changes in completion time are truly attributed to the limits of the testing environment. The results for speedup and efficiency are not as powerful as they may be but as more linguistic testing functions are included into the program, the scale of these results may improve.

**File Processing Speeds**

| | | | | |
|---|---|---|---|---|
| N=500 | (1, 0.04) | (2, 0.33) | (3, 0.34) | (4, 0.37) |
| N=750 | (1, 0.93) | (2, 0.72) | (3, 0.62) | (4, 0.63) |
| N=1,000 | (1, 1.42) | (2, 1.02) | (3, 0.88) | (4, 0.81) |
| N=1,500 | (1, 2.25) | (2, 1.49) | (3, 1.40) | (4, 1.31) |
| N=1,750 | (1, 3.16) | (2, 2.26) | (3, 1.88) | (4, 1.53) |

Table 3: Thread Completion Times by Seconds

**Line and Word Count for 1,750 Files**

| Number of processors | Line Count | Word Count |
|---|---|---|
| 2 | 139,655 | 4,380 |
| | 155,293 | 4,370 |
| 3 | 88,777 | 2,920 |
| | 99,859 | 2,920 |
| | 106,312 | 29,10 |
| 4 | 64,746 | 2,190 |
| | 73,460 | 2,190 |
| | 74,909 | 2,190 |
| | 81,833 | 2,180 |

Table 4: Word and Line Processing for 1,750 Files

**Speedup and Efficiency**

| $N$ | Speedup | | Efficiency |
|---|---|---|---|
| 500 | $\frac{1.30}{0.75}$ | **1.73** | $\frac{1.73}{4}$ **0.43** |
| 750 | $\frac{1.03}{0.98}$ | **1.66** | $\frac{1.66}{4}$ **0.415** |
| 1,000 | $\frac{1.99}{1.15}$ | **1.73** | $\frac{1.73}{4}$ **0.43** |
| 1,500 | $\frac{2.88}{1.56}$ | **1.88** | $\frac{1.88}{4}$ **0.47** |
| 1,750 | $\frac{19.07}{19.54}$ | **0.975** | $\frac{0.975}{4}$ **0.24** |

Table 5: Speedup and Efficiency of Program

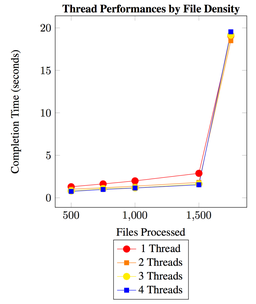
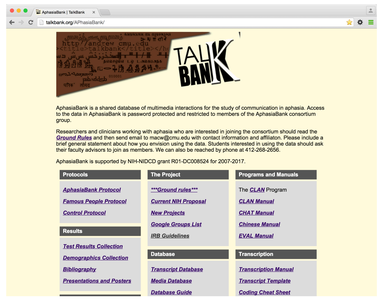



Figure 2: Thread Performance based on Time and Processing Volumes

## FUTURE WORK

AphasiaBank has already been funded for growth in other languages, including Madarin and Spanish, which have different prosodies. Implementing similar analyses in Spanish will provide greater insights towards prosody of individuals with non-fluent aphasia. It will be important to investigate the prosody of speech among Spanish (syllable-timed) and English (stress-timed) speakers. Analysis of different languages could provide more generalizable insights regarding lesion location and its impact on language processing [4,6].




## REFERENCES

1. Braber, N.., Patterson, K., Ellis, K., and Ralph, M. A. L. The relationship between phonological and morphological deficits in broca's aphasia: Further evidence from errors in verb inflection. *Brain and Language 92, 3* (2005), 278-287.

2. Collins, C. Docuburst: Document Content Visualization Using Language Structure. *Proceedings of IEEE Symposium on Information Visualization, Poster Compendium* (2006).

3. Galliers, J., Wilson, S., Roper, A., Cocks, N., Marshall, J., Muscroft, S., and Print, T. Words are not enough: Empowering people with aphasia in the design process. *In proceedings of the 12th Participatory Design Conference: Research Papers- Volume 1* (2012), ACM pp.51-60.

4. Kambanaros, M., Ibanescu, G., en Pescariu, S.Investigating Grammatical Word Class Distinctions in Bilingual Aphasic Individuals. *Nova Science Publishers*, (2009), 1–59.

5. MacWhinney, B., Fromm, D., Forbes, M. & Holland, A. AphasiaBank: Methods for studying discourse. *Aphasiology, 25,* (2011), 1286-1307.

6. Romani, C., Galluzzi, C., Bureca, I., en Olson, A.Effects of syllable structure in aphasic errors: Implications for a new model of speech production. *Cognitive Psychology 62*, 2 (2011), 151–192.

7. Roper, A. Accessibility of computer therapy and technology for people with aphasia. *ACM SIGACCESS Accessibility and Computing, 108* (2014), 50-53.

8. Viegas, F.B., Wattenberg, M., and Feinberg, J. Participatory visualization with Wordle. *Visualization and Computer Graphics, IEEE Transactions on 15.6* (2009): 1137-1144.